POWER-SAFE DATA MANAGEMENT SYSTEM

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] The present application claims the benefit of priority under 35 U.S.C. §120 as a continuation of U.S. patent application Ser. No. 13/073,588 entitled "Power-Safe Data Management System," filed on Mar. 28, 2011, the disclosure of which is hereby incorporated by reference in its entirety for all purposes.

BACKGROUND

[0002] Solid-state storage drives allow for very fast read-write times to a solid-state storage media. Processing speeds on a controller in the solid-state drive can therefore have a noticeable impact on read-write throughput. Solid-state drives must also maintain data coherency within a solid-state storage system. If the solid-state storage system loses power suddenly, it must be able to recover to a usable state and be able to maintain, for example, which portions of memory are safe to write to and which are currently being written by write processes.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] Systems and methods which embody the various features of the invention will now be described with reference to the following drawings, in which:

[0004] FIG. **1** shows an overview of a non-volatile storage device.

[0005] FIGS. **2**A, **3**A and **3**B show the contents of a master ecord for recording system status according to embodiments.

[0006] FIGS. **2**B, **4**A and **4**B show flowcharts for managing a master record according to embodiments.

[0007] FIGS. **5**A and **5**B show flowcharts for recovering system status from a master record stored to non-volatile storage.

DETAILED DESCRIPTION

[0008] While certain embodiments of the inventions are described, these embodiments are presented by way of example only, and are not intended to limit the scope of the inventions. Indeed, the novel methods and systems described herein may be embodied in a variety of other forms, Furthermore, various omissions, substitutions and changes in the form of the methods and systems described herein may be made without departing from the spirit of the inventions.

System Overview

[0009] Embodiments of the invention are directed to systems and methods for reducing read-write delays by a controller on a non-volatile storage system while maintaining a record of system activity in non-volatile memory for recovery purposes. In embodiments, the record of system activity can be used to restore system status and maintain coherency in the event of a power loss.

[0010] As used in this application, "non-volatile memory" typically refers to solid-state memory such as NAND flash. However, the systems and methods of this disclosure may also be useful in more conventional hard drives and hybrid drives including both solid-state and hard drive components. As such, while certain internal operations are referred to which typically are associated with solid-state drives, such as "wear leveling" and "garbage collection," analogous operations for hard drives can also take advantage of this disclosure.

[0011] In this disclosure, the term "superblock" is defined as a group of addresses on non-volatile storage media. This grouping of addresses can refer broadly to any method of grouping addresses together (e.g., page, blocks, etc.). In some embodiments, these groupings can include at least one block of write addresses from a plurality of dies in non-volatile storage media. In some embodiments, these groupings can be used by a controller to efficiently organize data access operations and manage the non-volatile storage system.

[0012] Referring to FIG. **1**, a system diagram shows a non-volatile storage device **100** in communication with a host device **130** according to an embodiment. As depicted in this embodiment, a non-volatile storage device **100** can include non-volatile memory arrays **120** and a controller **110**. The controller **110** is in communication with the non-volatile memory arrays **120** and with the host device **130**. The controller in this embodiment includes a user data manager **111**, a master record **112**, an internal operations manager **113**, and a media access request executor **114**. The user data manager **111** is in communication with a device driver **140** on the host device **130** and can receive and process memory requests from a device driver **140** on the host device. The internal operations manager **113** can include system clean up and maintenance tasks, such as garbage collection and wear leveling operations for maintaining the non-volatile memory arrays **120**, The arrays **120** may comprise a plurality of solid-state storage devices coupled to the controller **110**. The arrays may comprise, for example, flash integrated circuits, Chalcogenide RAM (C-RAM), Phase Change Memory (PC-RAM or PRAM), Programmable Metallization Cell RAM (PMC-RAM or PMCm), Ovonic Unified Memory (OUM), Resistance RAM (RRAM), NAND memory, NOR memory, EEPROM, Ferroelectric Memory (FeRAM), or other discrete NVM (non-volatile memory) chips. The solid-state storage devices may be physically divided into planes, blocks, pages, and sectors, as is known in the art. Other forms of storage (e.g., battery backed-up volatile DRAM or SRAM devices, magnetic disk drives, etc.) may additionally or alternatively be used.

[0013] Media access requests can be sent from the user data manager **111** or internal operations manager **113** to the media access request executor **114** for execution on non-volatile memory arrays **120**. In one embodiment, the media access request executor **114** schedules command execution on the non-volatile memory arrays and is responsible for any remaining translation tasks to execute a command on blocks of memory in the non-volatile memory arrays **120**. When the user data manager **111** or internal operations manager **113** perform a write command, the write command may be executed on a superblock designated for the next writes for the write command. Each superblock may designate a selection of blocks from a plurality of physical dies on the non-volatile memory array for write commands. During operation of the controller **110**, each write entity, such as the user data manager **111** and internal operations manager **113**, may have associated with it a running process that is assigned a superblock. As each process completes writing to